

Colouring Reconfiguration Is Fixed-Parameter Tractable ^{*}

Matthew Johnson¹, Dieter Kratsch², Stefan Kratsch³,
Viresh Patel⁴, and Daniël Paulusma¹

¹ School of Engineering and Computing Sciences, Durham University,
Science Laboratories, South Road, Durham DH1 3LE, United Kingdom
{matthew.johnson2,daniel.paulusma}@durham.ac.uk

² Laboratoire d'Informatique Théorique et Appliquée, Université de Lorraine,
57045 Metz Cedex 01, France, dieter.kratsch@univ-lorraine.fr

³ Institut für Softwaretechnik und Theoretische Informatik,
Technische Universität Berlin, Germany, stefan.kratsch@tu-berlin.de

⁴ School of Mathematical Sciences, Queen Mary, University of London,
Mile End Road, London E1 4NS, United Kingdom viresh.patel@qmul.ac.uk

Abstract. We prove that the problem of determining whether there exists a path of length at most ℓ between two given k -colourings in the reconfiguration graph for k -COLOURING is fixed-parameter tractable for all fixed $k \geq 1$, when parameterized by ℓ . This addresses an open problem of Mouawad, Nishimura, Raman, Simjour and Suzuki. We also show that the problem is polynomial-time solvable for $k = 3$, which solves an open problem of Cereceda, van den Heuvel and Johnson, and that it has no polynomial kernel for all $k \geq 4$, when parameterized by ℓ , unless the polynomial hierarchy collapses.

1 Introduction

Graph colouring has its origin in a nineteenth century map colouring problem and has now been an active area of research for more than 150 years. It has many applications within and beyond Computer Science and Mathematics. The goal is to find a mapping from the vertex set of the input graph to $\{1, 2, \dots, k\}$, the set of *colours*, such that adjacent vertices are not coloured alike and the number of colours k is minimized. Formally, the decision problem COLOURING asks whether such a k -colouring exists for a given graph G and integer k . If k is fixed, that is, not part of the input, the problem obtained is called k -COLOURING. The 3-COLOURING problem was an early example of a problem shown to be NP-complete [24], and it follows that COLOURING and k -COLOURING, for $k > 3$, are similarly intractable.

In this paper we consider the *reconfiguration graph* of instances of the k -COLOURING problem: this is a graph that has as its vertex set the set of all

^{*} This work has been supported by EPSRC (EP/G043434/1), by a Scheme 7 grant from the London Mathematical Society, and by the German Research Foundation (KR 4286/1).

possible proper k -colourings and that has an edge between vertices that represent colourings that differ on only a single vertex. The reconfiguration graph can be defined for any search problem: the vertices correspond to all solutions to the problem and the edges are defined by a symmetric adjacency relation on the solutions. This adjacency relation is normally chosen to represent a smallest possible change in the solution. The structure of the solution graph is, for many combinatorial decision problems, believed to be related to algorithmic performance. This was first noted for the SATISFIABILITY problem where a dichotomy was found [15]: for some cases (including all those for which SATISFIABILITY has polynomial time algorithms), the problem of finding a “path” between a pair of given solutions is in \mathbf{P} and the solution graph has linear diameter, while for all other cases, the problem is PSPACE-complete⁵ and the solution graph has exponential diameter. This has led, within the past five years, to research into the structure of solution graphs, not only for k -COLOURING [2,3,7,9,10,11] but also for many other problems, such as INDEPENDENT SET [8,22], LIST EDGE COLOURING [18,20], $L(2,1)$ -LABELING [19], SHORTEST PATH [4,6,23], and SUBSET SUM [21]. For more background information and motivation we refer to the recent survey of van den Heuvel [16].

We define a general version of the reconfiguration problem.

RECONFIGURATION

Instance: A graph $G = (V, E)$, two solutions α and β and a positive integer ℓ .

Question: Is there a path in the reconfiguration graph of G between α and β of length at most ℓ ?

Notice that the problem asks about the distance between solutions whereas the problem considered in some of the previous papers asks only whether a path between solutions exists.

One can view the input of a decision problem, such as the one defined above, as a pair (I, p) where I is the *main part* and p is the *parameter*, where the choice of parameter will depend on the structure of the problem. Then the problem is called *fixed-parameter tractable* (FPT) if any instance (I, p) can be solved in time $f(p)|I|^{O(1)}$ where f is a computable function that only depends on p (see the textbook of Niedermeier [27] for an overview). The paper of Mouawad, Nishimura, Raman, Simjour and Suzuki [26] was the first to consider the parameterized complexity of reconfiguration problems focusing on two natural parameters: the distance between solutions ℓ and the solution size s . They showed that RECONFIGURATION is fixed-parameter tractable for VERTEX COVER, BOUNDED HITTING SET and FEEDBACK VERTEX SET, when parameterized by s . They also proved that RECONFIGURATION is $\mathbf{W}[2]$ -hard for UNBOUNDED HITTING SET when parameterized by $s + \ell$, and they presented a list of maximization problems for which RECONFIGURATION is $\mathbf{W}[1]$ -hard when parameterized by $s + \ell$. Moreover, they showed that the deletion variant of every one of these maximization problems is $\mathbf{W}[1]$ -hard when parameterized by ℓ . Due to these \mathbf{W} -hardness

⁵ PSPACE-completeness appears to be the default complexity for intractable instances of this kind of problem; see [17].

results it is unlikely that there exist FPT algorithms for these problems when parameterized by ℓ , and they asked the following question:

Does there exist an NP-hard decision problem for which RECONFIGURATION is fixed-parameter tractable when parameterized by ℓ ?

In a recent paper, Mouawad, Nishimura and Raman [25] showed that RECONFIGURATION with parameter ℓ is W[1]-hard for VERTEX COVER restricted to bipartite graphs, but fixed-parameter tractable when restricted to graphs of bounded degree. The latter can be seen as an affirmative answer to the question in [26], since VERTEX COVER remains NP-complete for graphs of maximum degree at most 3 [14]. The question remained however, whether there is an affirmative answer for a problem where the input graphs are not restricted to some graph class.

Our Results In Section 2, we consider the k -COLOURING RECONFIGURATION problem. We answer the question of Mouawad et al. [26] for general problem instances by showing that RECONFIGURATION is fixed-parameter tractable for k -COLOURING for any fixed $k \geq 1$, when parameterized by ℓ . In fact we prove a stronger result, namely that the problem is fixed-parameter tractable when parameterized by $k + \ell$. As Bonsma and Cereceda [7] proved that the problem is PSPACE-complete for every fixed $k \geq 4$ even for bipartite graphs, we cannot hope for fixed-parameter tractability if k is the parameter.

It is well known [27] that a problem is FPT with respect to a parameter p if and only if it can be *kernelized*, that is, if and only if, for any problem instance (I, p) , it is possible to compute in polynomial time an equivalent problem instance (I', p') with $p' \leq p$ and $|I'| \leq g(p)$ for some computable function g (two problem instances are equivalent if and only if they are both yes-instances or both no-instances). If $g(p)$ is a polynomial, then the problem is said to have a polynomial kernel. Hence, it is natural to ask whether RECONFIGURATION for k -COLOURING has a polynomial kernel when parameterized by ℓ . In Section 3 we show that the problem is polynomial-time solvable for $k = 3$, thereby solving the open problem of Cereceda, van den Heuvel and Johnson [11]. However, for all $k \geq 4$, we show in Section 4 that the problem has no polynomial kernel (up to the standard complexity assumption that $\text{NP} \not\subseteq \text{coNP/poly}$). In fact we show that the problem has no polynomial kernel for all $k \geq 4$ even when restricted to inputs where the two proper k -colourings of G differ in only two vertices. This result is tight as the problem becomes trivial if the two given k -colourings differ in only one vertex. In particular, it is the first result that addresses the question whether a reconfiguration problem admits a polynomial kernelization.

2 An FPT Algorithm for k -Colouring Reconfiguration

Let $G = (V, E)$ be a graph on n vertices. A *colouring* of G is a mapping $c : V \rightarrow \{1, 2, \dots\}$. If $c(u) \neq c(v)$ whenever $uv \in E$ then c is a *proper* colouring. We call $c(u)$ the *colour* of u . Let $k \geq 1$ be an integer. A (*proper*) k -colouring of G is a

(proper) colouring with $1 \leq c(u) \leq k$ for all $u \in V$. We refer to the textbook of Diestel [12] for any undefined graph-theoretic terms.

We consider the following problem:

k -COLOURING RECONFIGURATION

Instance : A graph $G = (V, E)$, two k -colourings α and β and a positive integer ℓ .

Question : Is there a path in the reconfiguration graph of G between α and β of length at most ℓ ?

We will describe an FPT algorithm for k -COLOURING RECONFIGURATION when parameterized by ℓ . First we prove a number of lemmas concerning the vertices that might be recoloured if a path between α and β of length at most ℓ does exist. That is, we assume that (G, α, β, ℓ) is a yes-instance of k -COLOURING RECONFIGURATION.

For any two colourings c and d , we say that c and d *agree* on a vertex u if $c(u) = d(u)$ and that otherwise they *disagree* on u .

An $(\alpha \rightarrow \beta)$ -recolouring R of length ℓ is a sequence of proper colourings c_0, \dots, c_ℓ where $c_0 = \alpha$ and $c_\ell = \beta$, and, for $1 \leq q \leq \ell$, c_q and c_{q-1} disagree on at most one vertex. So possibly $c_q = c_{q+1}$ though in this case c_q could be deleted and the sequence that remained would also be an $(\alpha \rightarrow \beta)$ -recolouring. The set $\{c_q c_{q+1} : c_q \neq c_{q+1}\}$ is a set of edges in the reconfiguration graph that corresponds to a walk from α to β .

From now on, assume that $R = c_0, \dots, c_\ell$ is an $(\alpha \rightarrow \beta)$ -recolouring of G of *minimum length*.

We say that R *recolours* a vertex u if $c_q(u) \neq \alpha(u)$ for some q . Notice that if for each recoloured vertex u we find the least q such that $c_q(u) \neq \alpha(u)$, these values must be distinct (else c_q and c_{q-1} disagree on more than one vertex). Thus the number of distinct vertices recoloured by R is at most ℓ . We will prove something stronger. For $0 \leq q \leq \ell$, let W_q be the set of vertices on which c_0 and c_q disagree, that is,

$$W_q = \{u \in V : c_0(u) \neq c_q(u)\}.$$

Lemma 1. *For all q with $1 \leq q \leq \ell$, the set W_q has size $|W_q| \leq q$.*

Proof. Suppose this is false and let r be the smallest value such that $|W_r| > r$. So $|W_{r-1}| \leq r-1$ (clearly $r-1 \geq 0$ as W_0 is the empty set). Then there are (at least) two vertices v_1, v_2 in $W_r \setminus W_{r-1}$, and so, for $i \in \{1, 2\}$, $c_{r-1}(v_i) = c_0(v_i) \neq c_r(v_i)$, and c_r and c_{r-1} disagree on more than one vertex. This contradiction proves the lemma. \square

For any $u \in V$, let $N(u)$ be the set of neighbours of u . For any $v \in N(u)$, let $N(u, v) = \{w \in N(u) : \alpha(w) = \alpha(v)\}$; that is, the set of neighbours of u with the same colour as v in α .

Let $A_0 = \{v \in V : \alpha(v) \neq \beta(v)\}$ be the set of vertices on which α and β disagree. For $i \geq 1$, let

$$A_i = \bigcup_{u \in A_{i-1}} \{v \in N(u) : |N(u, v)| \leq \ell\}.$$

That is, to find A_{i+1} consider each vertex u in A_i and partition $N(u)$ into colour classes (according to the colouring α). Vertices in $N(u)$ that belong to colour classes of size at most ℓ belong to A_{i+1} . Note that two sets A_h and A_i need not be disjoint, and let

$$A = \bigcup_i A_i.$$

Our goal is first to show that each vertex recoloured by R must be in A . We will then show how to find a subset A^* of A with size bounded by a function of $k + \ell$ such that that R only recolours vertices from A^* and in fact only recolours ℓ of them. This will then enable us to use brute-force to find R or some other $(\alpha \rightarrow \beta)$ -recolouring of G (if it exists).

Lemma 2. *Each vertex recoloured by R belongs to A .*

Proof. For $0 \leq q \leq \ell$, let d_q be a (not necessarily proper) colouring of G such that

- if $u \in A$, $d_q(u) = c_q(u)$;
- if $u \notin A$, $d_q(u) = \alpha(u)$.

Let R' be the sequence d_0, \dots, d_ℓ . We notice that as $d_0(u)$ is either $c_0(u)$ or $\alpha(u)$, and $c_0 = \alpha$, we have $d_0 = \alpha$. If $u \in A$, $d_\ell(u) = c_\ell(u) = \beta(u)$, and if $u \notin A$, $d_\ell(u) = \alpha(u) = \beta(u)$ (since α and β only disagree on vertices in $A_0 \subseteq A$); thus $d_\ell = \beta$.

If we can show that R' contains only proper colourings, then we will have shown that it is an $(\alpha \rightarrow \beta)$ -recolouring.

Suppose instead that R' contains a colouring d_q such that there is an edge uv with $d_q(u) = d_q(v)$. If u and v both belong to A , then we would have $c_q(u) = c_q(v)$, and if neither belong to A , then $\alpha(u) = \alpha(v)$. Since we know that c_q and α are proper colourings we can assume, without loss of generality, that $u \in A$ and $v \notin A$.

So $c_q(u) = d_q(u) = d_q(v) = \alpha(v)$ by the definition of d_q . As $u \in A$, it belongs to A_i for some i . As $v \in N(u)$, if $|N(u, v)| \leq \ell$, then v would belong to A_{i+1} . Hence, as $v \notin A$, we know $|N(u, v)| > \ell$. As $c_q(u) = \alpha(v)$ and c_q is proper, for each $w \in N(u, v)$, $c_q(w) \neq c_q(u) = \alpha(v) = \alpha(w)$. Thus $W_q \supseteq N(u, v)$ and so $|W_q| > \ell \geq q$. This contradiction of Lemma 1 tells us that each d_q is a proper colouring, and R' is an $(\alpha \rightarrow \beta)$ -recolouring of length ℓ .

To complete the proof: if R does recolour a vertex $v \notin A$, then there is a pair of colourings c_q and c_{q+1} that differ only on v . But then d_q and d_{q+1} are identical colourings and if we remove d_q from R' the sequence of colourings that remains is also an $(\alpha \rightarrow \beta)$ -recolouring, which has length shorter than ℓ , contradicting that R has minimum length. The lemma is proved. \square

By Lemma 2 we can assume that R is a minimum length $(\alpha \rightarrow \beta)$ -recolouring that recolours only vertices in A . We need something stronger than this though: to show that R in fact only recolours vertices in $A_0, \dots, A_{\ell-1}$. The next three

lemmas lead us towards that conclusion, but first some more definitions are needed.

Let z be the greatest value such that R recolours a vertex which is in A_z but not in A_i for any $i < z$, and let

$$x \in A_z \setminus \bigcup_{i < z} A_i$$

be a vertex recoloured by R . Let $B_0 = \{x\}$. For $j \geq 1$, let

$$B_j = \bigcup_{u \in B_{j-1}} \{v \in N(u), \exists q : c_q(v) = \alpha(u)\}.$$

That is, if u is in B_{j-1} , then any neighbour v of u with $c_q(v) = \alpha(u)$, for some q , belongs to B_j . Let

$$B = \bigcup_j B_j.$$

Lemma 3. *Each vertex in B is recoloured by R .*

Proof. This is true for x (the only vertex in B_0) by the definition of x . Now let $j > 0$, and let v be a vertex in B_j . Then there is some colouring c_q and vertex $u \in N(v) \cap B_{j-1}$ where $c_q(v) = \alpha(u) \neq \alpha(v)$ (the latter inequality holds because α is a proper k -colouring). Hence R recolours v . \square

Lemma 4. *The intersection of A_0 and B is not empty.*

Proof. To obtain a contradiction, suppose that no vertex of A_0 belongs to B . For $0 \leq q \leq \ell$, let e_q be a (not necessarily proper) colouring of G such that

- if $u \notin B$, $e_q(u) = c_q(u)$;
- if $u \in B$, $e_q(u) = \alpha(u)$.

Let R'' be the sequence e_0, \dots, e_ℓ . It can be seen that $e_0 = \alpha$ and $e_\ell = \beta$ (remembering, for the latter, that we are assuming that each $u \in B$ is not in A_0 , so $e_\ell(u) = \alpha(u) = \beta(u)$). If we can show that R'' contains only proper colourings, then we will have shown that it is an $(\alpha \rightarrow \beta)$ -recolouring.

Suppose instead that R'' contains a colouring e_q such that there is an edge uv with $e_q(u) = e_q(v)$. If u and v both belong to B , then we would have $\alpha(u) = \alpha(v)$, and if neither belong to B , then $c_q(u) = c_q(v)$. However, since we know that c_q and α are proper colourings we can assume, without loss of generality, that $u \in B$ and $v \notin B$.

As $u \in B$, we have $u \in B_j$ for some j . As $e_q(u) = \alpha(u)$ and $v \notin B$, we also have that $c_q(v) = e_q(v) = e_q(u) = \alpha(u)$. But this tells us that $v \in B_{j+1} \subseteq B$. From this contradiction, we conclude that R'' is, in fact, an $(\alpha \rightarrow \beta)$ -recolouring of length ℓ .

Finally we note that $x \in B$ is recoloured by R (any vertex in B would suffice here, but x is one we know exists). Hence there is some q such that

$c_q(x) \neq c_{q+1}(x)$. But $e_q(x) = e_{q+1}(x) = \alpha(x)$ so e_q and e_{q+1} are identical. Thus if we delete e_q from R'' , we are left with another $(\alpha \rightarrow \beta)$ -recolouring, but this one has length shorter than ℓ . As R was chosen to have minimum length, we have a contradiction which proves the lemma. \square

For $j \geq 0$, let $m(B_j)$ be the least i such that B_j contains a vertex in A_i , that is,

$$m(B_j) = \min_i \{B_j \cap A_i \neq \emptyset\}.$$

Lemma 5. *For all $j \geq 0$, $m(B_{j+1}) \geq m(B_j) - 1$.*

Proof. Suppose that, on the contrary, there exists an index j is such that

$$m(B_{j+1}) = i < m(B_j) - 1.$$

Then there exists a vertex $v \in B_{j+1} \cap A_i$ while $m(B_j) \geq i + 2$, so B_j contains no vertex of A_h for any $h < i + 2$.

Because $v \in B_{j+1}$, there is a q such that $c_q(v) = \alpha(u)$ for some $u \in B_j$ (by definition), where $u \in N(v)$. As B_j contains no vertex of A_h for any $h < i + 2$, we find that $u \notin A_{i+1}$. Hence, we have $|N(v, u)| > \ell$. As c_q is proper, for each $w \in N(v, u)$, $c_q(w) \neq c_q(v) = \alpha(u) = \alpha(w)$. Thus $W_q \supseteq N(v, u)$ and hence $|W_q| > \ell \geq q$. This is a contradiction of Lemma 1. Hence, we have proven the lemma. \square

Lemmas 3–5 enable us to prove the following statement.

Lemma 6. *For $0 \leq i \leq z$, there is a vertex in $A_i \setminus (\bigcup_{h < i} A_h)$ that is recoloured by R .*

Proof. As $B_0 = \{x\}$ and $x \in A_z$, we have $m(B_0) = z$. By Lemma 4, there is some p such that B_p intersects A_0 , so $m(B_p) = 0$. Suppose there is an integer i , $0 < i < z$, such that there does not exist a set B_j with $m(B_j) = i$. Then choose r as small as possible such that $m(B_r) \leq i - 1$ (there certainly is such an r since $m(B_p) = 0 \leq i - 1$). Then, as $m(B_{r-1}) > i$ by the choice of r and our assumption on i , we find that $m(B_r) \leq i - 1 < m(B_{r-1}) - 1$, which contradicts Lemma 5.

So for each i , $0 \leq i \leq z$, there exists a set B_j with $m(B_j) = i$. By the definition of $m(B_j)$, there is a vertex of B_j that belongs to $A_i \setminus (\bigcup_{h < i} A_h)$, and, by Lemma 3, every vertex in B is recoloured by R . \square

We are finally able to prove the following crucial lemma.

Lemma 7. *Each vertex recoloured by R belongs to the union of $A_0, \dots, A_{\ell-1}$.*

Proof. Recall that z is the greatest value such that R recolours a vertex that is in A_z but not in A_i for any $i < z$. Hence, each vertex recoloured by R belongs to the union of A_0, \dots, A_z . We must prove that $z < \ell$.

By Lemma 6, there is a vertex $v_i \in A_i \setminus \bigcup_{h < i} A_h$ recoloured by R , $0 \leq i \leq z$. Because v_0, \dots, v_z are distinct vertices, we have $z + 1$ vertices that are recoloured by R . Because R has length ℓ , it can recolour at most ℓ vertices. We conclude that $z + 1 \leq \ell$. This completes the proof. \square

By Lemma 7, we find that all vertices that are recoloured by R belong to the set

$$A^* = \bigcup_{i=0}^{\ell-1} A_i.$$

We now show that the size of A^* is bounded by a function that depends only on k and ℓ .

Lemma 8. *The set A^* has size $|A^*| \leq \ell \cdot (k\ell)^\ell$.*

Proof. Recall that A_0 is the set of vertices on which α and β disagree. Hence, all vertices of A_0 need to be recoloured by R . Thus we have $|A_0| \leq \ell$.

Now let i be such that $1 \leq i \leq \ell - 1$. Recall that the set A_i is defined as $A_i = \bigcup_{u \in A_{i-1}} \{v \in N(u) : |N(u, v)| \leq \ell\}$. By this recursive definition, each vertex of A_i is a neighbour of a vertex of A_{i-1} , and each vertex of A_{i-1} has at most $k \cdot \ell$ neighbours in A_i . Consequently $|A_i| \leq |A_{i-1}| \cdot k \cdot \ell$, for all $1 \leq i \leq \ell - 1$. The achieved upper bounds imply

$$|A^*| = \sum_{i=0}^{\ell-1} |A_i| \leq \sum_{i=0}^{\ell-1} \ell \cdot (k \cdot \ell)^i \leq \ell \cdot \frac{(k \cdot \ell)^\ell - 1}{k \cdot \ell - 1} \leq \ell \cdot (k \cdot \ell)^\ell.$$

□

We are now ready to present our FPT algorithm for solving k -COLOURING RECONFIGURATION with parameter $k + \ell$.

Theorem 1. *The k -COLOURING RECONFIGURATION problem can be solved in time $O((k \cdot \ell)^{\ell^2 + \ell} \cdot \ell n^2)$.*

Proof. Let $k \geq 1$, and let (G, α, β, ℓ) be an instance of k -COLOURING RECONFIGURATION, where G is a graph on n vertices, and α, β are two proper k -colourings of G .

Our algorithm first computes the set A^* in $O(n^2)$ time. By Lemma 8, we find that $|A^*| \leq \ell \cdot (k\ell)^\ell$.

By Lemma 7, our algorithm only has to search for a “path” of length at most ℓ in the reconfiguration graph among the vertices of A^* . Note that by allowing consecutive recolourings to be equal we may restrict our search to $(\alpha \rightarrow \beta)$ -recolourings of length exactly ℓ . Below we explain how the search is done and how much time it takes.

Our algorithm uses brute force to enumerate all possible sequences of pairs (v_i, c_i) , such that for all $0 \leq i \leq \ell - 1$, v_i is a vertex in A^* and c_i is a colour in $\{1, \dots, k\}$. For each such sequence the algorithm does as follows. Starting from α , it recolours v_i with colour c_i for $i = 0, \dots, \ell - 1$. As soon as this results in a k -colouring that is not proper, it stops considering the sequence. If not, it checks whether the resulting colouring is equal to β . If this happens, then there is a path of length ℓ in the reconfiguration graph. Hence, the algorithm outputs **yes**. Otherwise, that is, if no sequence has this property, it outputs **no**.

Processing one sequence takes time $O(\ell n^2)$. By using Lemma 8, we find that the number of sequences is at most

$$(|A^*| \cdot k)^\ell \leq ((\ell \cdot (k \cdot \ell)^\ell) \cdot k)^\ell \leq (k \cdot \ell)^{\ell^2 + \ell}.$$

Thus the running time of the algorithm is $O((k \cdot \ell)^{\ell^2 + \ell} \cdot \ell n^2)$. This completes the proof. \square

We observe that it is easy to use A^* and its upper bound, given in Lemma 8, to construct an (exponential size) kernel.

3 A Polynomial-Time Algorithm for $k = 3$

Cereceda, van den Heuvel and Johnson [11] showed that, for a given graph G with two given proper 3-colourings α and β , it can be decided in polynomial time whether there exists an $(\alpha \rightarrow \beta)$ -recolouring of G . Their algorithm is guaranteed to yield an $(\alpha \rightarrow \beta)$ -recolouring of shortest length only for restricted instances. In this section we complete their study.

Let G be a graph with a proper 3-colouring α . Let $F_\alpha(G)$ be the set of those vertices in G that are *fixed*, that is, a vertex v is in $F_\alpha(G)$ if and only if, for any proper 3-colouring γ with $\gamma(v) \neq \alpha(v)$, there is no $(\alpha \rightarrow \gamma)$ -recolouring of G .

The following result, which we use as a lemma, is due to Cereceda et al. [11].

Lemma 9 ([11]). *The 3-COLOURING RECONFIGURATION problem can be solved in polynomial time on instances (G, α, β, ℓ) with $F_\alpha(G) \neq \emptyset$.*

For a graph G , let $F(G)$ consist of those vertices, two of whose neighbours are adjacent to each other.

Lemma 10. *Let G be a graph. Then $F(G) \subseteq F_\alpha(G)$ for every proper 3-colouring α .*

We are now ready to prove our result.

Theorem 2. *The 3-COLOURING RECONFIGURATION can be solved in polynomial time.*

Proof. Let (G, α, β, ℓ) be an instance of 3-COLOURING RECONFIGURATION. We may assume without loss of generality that $\alpha \neq \beta$. Let u be a vertex with $\alpha(u) \neq \beta(u)$, say $\alpha(u) = 1$ and $\beta(u) = 2$.

From G we construct a new graph as follows. Let $r \geq 1$ be an integer (the meaning of which will become clear later) and take r copies G_1, \dots, G_r of G . Denote the copy of u in G_j by u_j for $j = 1, \dots, r$. Take a path $v_1 v_2 \dots v_{6\ell-3}$ on $6\ell - 3$ new vertices $v_1, \dots, v_{6\ell-3}$ and add an edge $u_j v_1$ for $j = 1, \dots, r$. Finally take a triangle xyz on three new vertices x, y, z and add the edge $v_{6\ell-3} x$. Call the resulting graph G_r^* . Note that $F(G_r^*) \neq \emptyset$, because $x, y, z \in F(G_r^*)$.

We first define a proper 3-colouring α' on G_r^* . We let α' coincide with α on each copy G_j of G ; in particular this means that $\alpha'(u_j) = \alpha(u) = 1$. We let

$\alpha'(v_j) = 2$ whenever j is odd and $\alpha'(v_j) = 1$ whenever j is even. We define $\alpha'(x) = 1$, $\alpha'(y) = 2$ and $\alpha'(z) = 3$. Note that α' is indeed a proper 3-colouring of G_r^* .

We now define ℓ other proper 3-colourings of G_r^* denoted $\beta'_1, \dots, \beta'_\ell$ as follows. Let $1 \leq i \leq \ell$. We let β'_i coincide with β on each copy G_j of G . For $h = 0, \dots, 2i - 2$, we set $\beta'_i(v_{3h+1}) = 3$, $\beta'_i(v_{3h+2}) = 1$ and $\beta'_i(v_{3h+3}) = 2$. If $i \leq \ell - 1$ then for $h = 6i - 2, \dots, 6\ell - 3$, we give the remaining vertices of the path colour $\beta'_i(v_h) = \alpha'_i(v_h)$. Finally, we let β'_i coincide with α' on $\{x, y, z\}$. Note that β'_i is indeed a proper 3-colouring of G_r^* .

We need the following claim.

Claim 1. If there exists an $(\alpha \rightarrow \beta)$ -recolouring of G that has length at most ℓ , then there exists a 3-colouring β'_i such that there exists an $(\alpha' \rightarrow \beta'_i)$ -recolouring of G_r^ that has length at most $\ell r + 4\ell^2$ for all $r \geq 1$.*

We prove Claim 1 as follows. Suppose there exists an $(\alpha \rightarrow \beta)$ -recolouring R of G that has length at most ℓ . We construct an $(\alpha' \rightarrow \beta'_i)$ -recolouring R^* of G_r^* as follows. Every time R recolours a vertex in G , we let R^* recolour the r copies in G_r^* to the same colour. This may lead to a 3-colouring of G_r^* that is not proper, but this only happens if R recolours u into the colour of v_1 . We avoid obtaining an improper 3-colouring by first letting R^* recolour v_1 before recolouring each u_i . We emphasise that we only recolour v_1 if it is necessary to do so. Before we can recolour v_1 we may first be required to recolour a number of vertices further along the path $v_1 \cdots v_{6\ell-3}$, depending on how many times we were forced to recolour v_1 already. Below we explain this in detail.

As R has length at most ℓ , we find that u , and hence each u_i , is recoloured at most ℓ times. The first time each u_i must be recoloured to the same colour of v_1 , each u_i must get colour 2, and we only need to recolour v_1 into colour 3. The second time this happens, each u_i must get colour 3, and we need to first recolour v_3 into 3, then v_2 into 2 and finally v_1 into 1, and so on. Using induction, this means that, in order to recolour u_i , we need at most

$$1 + 3 + \cdots + 2\ell - 1 \leq 4\ell^2$$

extra recolourings involving the first $2\ell - 1$ vertices from the path $v_1 v_2 \cdots v_{6\ell-3}$. Because the length of this path is $6\ell - 3 \geq 2\ell - 1$, we can do so without being forced to recolour x, y, z (which would not be possible). Because the final recolouring of each u_i changes the colour of u_i into 2, the final colours of the vertices $v_1, \dots, v_{6\ell-3}$ will be the colours prescribed by some β'_i , depending on how many times each u_i was recoloured to the colour of its neighbour v_1 . This means that the length of R^* is at most $\ell r + 4\ell^2$. Hence, Claim 1 is proven.

We are now ready to fix the value of r and we let

$$r = 4\ell^2 + 1.$$

Because $F(G_r^*) \neq \emptyset$, we find that $F_{\alpha'}(G_r^*) \neq \emptyset$ due to Lemma 10. This allows us to apply Lemma 9 on input $(G_r^*, \alpha', \beta'_i, \ell r + 4\ell^2)$ for $i = 1, \dots, \ell$. This will tell us

in polynomial time if there exists an $(\alpha' \rightarrow \beta'_i)$ -recolouring of G_r^* that has length at most $\ell r + 4\ell^2$. If not then we find that G has no $(\alpha \rightarrow \beta)$ -recolouring of length at most ℓ by Claim 1.

From now on suppose that for some $1 \leq i \leq \ell$ we have found an $(\alpha' \rightarrow \beta'_i)$ -recolouring R^* of G_r^* that has length at most $\ell r + 4\ell^2$. We will show that G has an $(\alpha \rightarrow \beta)$ -recolouring of length at most ℓ .

Because α' and β'_i coincide on each copy G_j , the restriction R_j of R^* to each G_j is an $(\alpha \rightarrow \beta)$ -recolouring of G_j , and thus of G . Let ℓ_j be the length of R_j .

Consider a copy G_h with minimum ℓ_h . Suppose that $\ell_h \geq \ell + 1$. Then R^* has length at least $r\ell_h \geq r(\ell + 1) = \ell r + r = \ell r + 4\ell^2 + 1$. This is not possible as R^* has length at most $\ell r + 4\ell^2$. Hence, $\ell_h \leq \ell$ and we conclude that R_h is an $(\alpha \rightarrow \beta)$ -recolouring of G that has length at most ℓ . This completes the proof of Theorem 2. \square

4 A Lower Bound for Kernelization for $k \geq 4$

We prove that, for all $k \geq 4$, the k -COLOURING RECONFIGURATION problem parameterized by ℓ , the number of permitted recolouring steps, does not admit a polynomial kernelization unless $\text{NP} \subseteq \text{coNP/poly}$. To prove the result we give a so-called polynomial parameter transformation from a problem that is known not to admit a polynomial kernelization, assuming $\text{NP} \not\subseteq \text{coNP/poly}$, and also no polynomial compression.⁶ A polynomial parameter transformation, short PPT, is a standard Karp reduction with the additional property that the parameter value of the returned instance is polynomially bounded in the parameter of the input instance. It is well known and easy to see that a PPT from a problem without polynomial compression implies that the target problem admits no polynomial compression and hence also no polynomial kernelization (cf. [1]).

As our source problem we use HITTING SET. This problem takes as input a finite set U , a set $\mathcal{F} \subseteq 2^U$ and an integer p and asks whether there exists a *hitting set* $S \subseteq U$ of size at most p , that is, a set S with $|S| \leq p$ such that every $F \in \mathcal{F}$ contains at least one element of S . The HITTING SET problem can also be formulated as the RED-BLUE DOMINATING SET problem, which takes a bipartite graph with partition classes R and B and asks whether there exists a set $D \subseteq R$ such that every vertex of B has at least one neighbour in D . Dom, Lokshtanov, and Saurabh [13] showed that the RED-BLUE DOMINATING SET problem, parameterized by $k + |B|$, does not admit a polynomial kernelization (unless $\text{NP} \subseteq \text{coNP/poly}$). As $k \leq |B|$ holds for any non-trivial instance, the same result holds with parameter $|B|$ instead of $k + |B|$. Since the result for RED-BLUE DOMINATING SET makes use of the standard framework of giving or/and-compositions, it is known to also rule out polynomial compressions (cf. [1]).

⁶ A (polynomial) compression is a relaxed form of (polynomial) kernelization: The output may be with respect to *any (possibly unparameterized) problem*.

Lemma 11. *The HITTING SET problem parameterized by $|\mathcal{F}|$ does not admit a polynomial compression unless $\text{NP} \not\subseteq \text{coNP}/\text{poly}$.*

We are now ready to prove our result.

Theorem 3. *Let $k \geq 4$. The k -COLOURING RECONFIGURATION problem parameterized by ℓ , the permitted number of recolourings, does not admit a polynomial kernelization (or polynomial compression) unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.*

Proof. By Lemma 11 it suffices to show that there is a polynomial parameter transformation from HITTING SET parameterized by $|\mathcal{F}|$ to k -COLOURING RECONFIGURATION parameterized by ℓ .

Let (U, \mathcal{F}, p) be an instance of HITTING SET. Let $m = |\mathcal{F}|$. We give a polynomial-time construction of an equivalent instance (G, α, β, ℓ) of the k -COLOURING RECONFIGURATION problem, where ℓ is polynomially bounded in m . Note that $p < m$ or else the instance (U, \mathcal{F}, p, m) is trivially yes and our transformation is trivial.

Construction. We begin with a standard argument for bounding the size of U : If any two elements of U occur exactly in the same sets of \mathcal{F} then we will never need both for a minimum hitting set and can safely discard either one of them. Thus, without loss of generality, we may assume that no two such elements exist, which implies that $|U| \leq 2^{|\mathcal{F}|} = 2^m$. Thus, our final parameter value ℓ may depend polynomially on $\log |U| = O(m)$. For convenience let $n = |U|$ and let $U = \{1, \dots, n\}$.

The graph G will consist of four components:

1. Two adjacent vertices s, t with $\alpha(s) = \beta(t) = 2$ and $\alpha(t) = \beta(s) = 3$. These are the only two vertices with different colours in α and β .
2. A clique of k vertices u_1, \dots, u_k with colours $\alpha(u_i) = \beta(u_i) = i$ that will be used to control permissible colours for all other vertices.
3. An independent set of vertices v_1, \dots, v_n , one for each element of U , each with colour $\alpha(v_i) = \beta(v_i) = 4$ that will be used to simulate selection of a hitting set of size at most p .
4. One *selection gadget* for each set $F \in \mathcal{F}$ that simulates a selection of one element of the hitting set to hit F . These will be described later as they are somewhat more involved.

Clearly, since each vertex of the clique is adjacent to all other colours but its own, it is impossible to recolour any vertex u_i and obtain a proper k -colouring. Thus, we can use adjacency to parts of the clique to forbid certain colours from being used for other vertices. Generally, all other vertices are made adjacent to all of u_5, \dots, u_k , effectively reducing the setting to the case that $k = 4$. (Mainly this ensures that our reduction works for all values $k \geq 4$.) Additionally, all vertices v_1, \dots, v_n are made adjacent to u_2 and u_3 , which in total allows only colours 1 and 4 to be used for the independent set.

Finally, we restrict vertex s to colours 2 and 3 and vertex t to colours 2, 3, and 4. Note that if only 2 and 3 were possible for both s and t , then it would be

impossible to recolour even just the graph on s and t . Using the additional option of colour 4 for t the following sequence works: (1) recolour t to 4, (2) recolour s to 3, and (3) recolour t to 2. The hitting set question will be encoded in a part of the graph that requires recolouring in order not to obstruct colouring t with colour 4 (and will be reverted once t is coloured 2).

We will now describe the construction of the selection gadgets. The basic building block is a claw on vertices $a_{\dagger}, b_{\dagger}, c_{\dagger}, d_{\dagger}$ with c_{\dagger} the center vertex (adjacent to $a_{\dagger}, b_{\dagger}, d_{\dagger}$) and with the following α and β colours and forbidden colours:

1. For a_{\dagger} we have $\alpha(a_{\dagger}) = \beta(a_{\dagger}) = 2$, and, using adjacency to the k -clique, only colours 2 and 4 allow proper k -colourings.
2. Similarly, for b_{\dagger} we have $\alpha(b_{\dagger}) = \beta(b_{\dagger}) = 3$, and only colours 3 and 4 are possible.
3. For the center vertex c_{\dagger} we have $\alpha(c_{\dagger}) = \beta(c_{\dagger}) = 1$, and only colours 1, 2, and 3 are possible.
4. For vertex d_{\dagger} we have $\alpha(d_{\dagger}) = \beta(d_{\dagger}) = 4$, and only colours 1 and 4 are possible.

The idea is that connecting such claws in a tree-like fashion gives the desired selection gadget. For the basic functionality that is to recolour d_{\dagger} with 1 it is necessary to first recolour c_{\dagger} to either 2 or 3. This in turn first requires a recolouring of (accordingly) either a_{\dagger} to 4 or b_{\dagger} to 4. Now if both a_{\dagger} and b_{\dagger} are adjacent to, say, d_{\ddagger} and d_{\ddagger}' of further such claws then the same argumentation continues since we again would need to recolour first d_{\ddagger} from 4 to 1 or d_{\ddagger}' from 4 to 1.

Now, let us describe the tree-like arrangement in more detail. For convenience, let us assume that $n = 2^r$ for some integer r , which we can achieve by adding at most $n - 1$ dummy elements to U that never occur in any set (thereby at most doubling n). We make copies of the claw construction that we just explained, for all values of

$$\dagger \in \{(F, x, y) \mid F \in \mathcal{F}, x \in \{0, \dots, r-1\}, y \in \{1, \dots, 2^x\}\}.$$

We connect these claws as follows (see Figure 1):

1. Each vertex $d_{F,0,1}$ is made adjacent to the vertex t .
2. Each vertex $d_{F,x,y}$, with $x \in \{1, \dots, r-1\}$ is made adjacent to $a_{F,x-1,(y+1)/2}$ if y is odd, and to $b_{F,x-1,y/2}$ if y is even.
3. Each vertex $a_{F,r-1,y}$ is made adjacent to vertex v_{2y-1} of the independent set.
4. Each vertex $b_{F,r-1,y}$ is made adjacent to vertex v_{2y} .

The idea is that to recolour $d_{F,0,1}$ to 1 it is ultimately necessary to first recolour some vertex v_j in the independent set from 4 to 1. This in turn allows to recolour the adjacent $a_{F,\cdot,\cdot}$ or $b_{F,\cdot,\cdot}$ vertex to 4 and then propagate possible recolourings towards $d_{F,0,1}$.

Note that so far we have made no distinction between trees made for different sets $F \in \mathcal{F}$; we now make the following modifications to vertices $a_{F,r-1,y}$

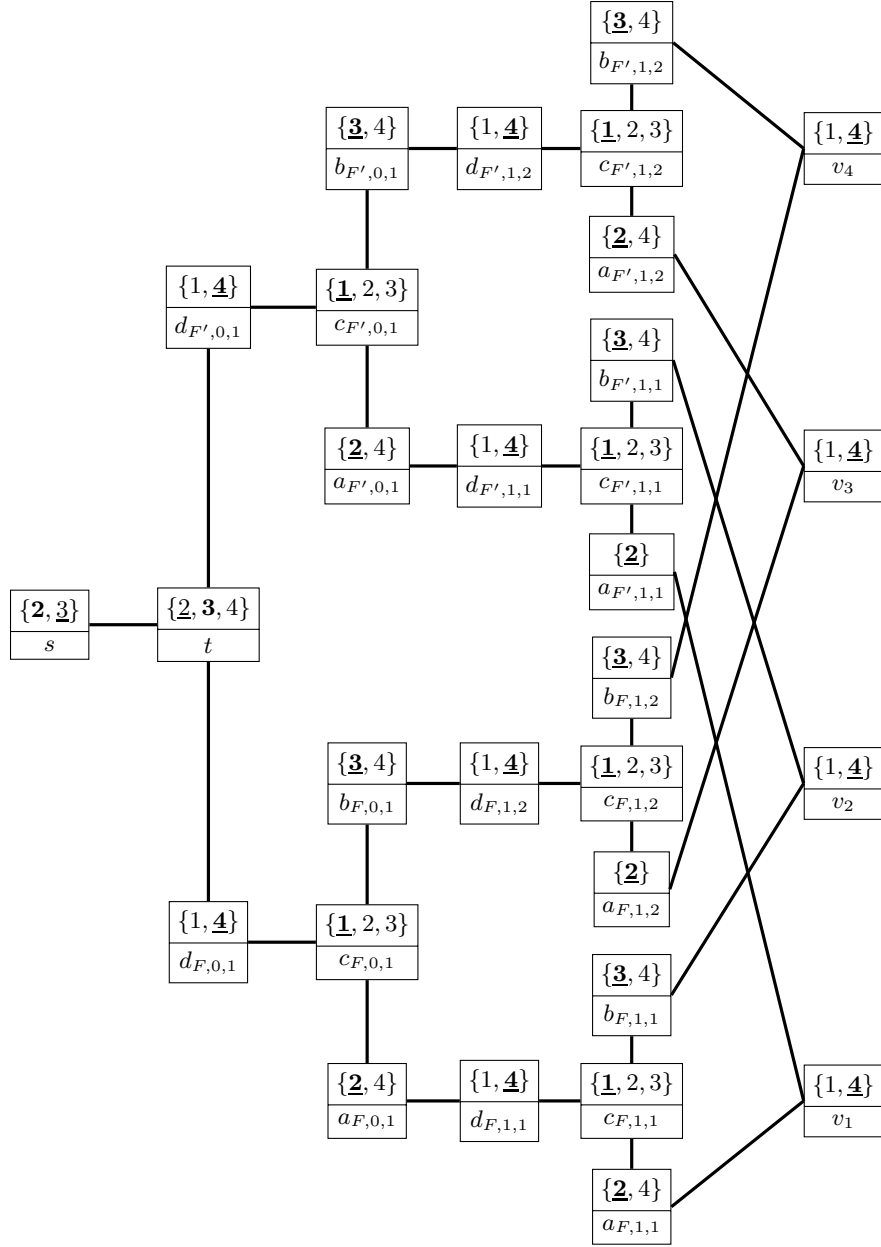


Fig. 1. A small example of the lower bound construction for (U, \mathcal{F}, p) and $k = 4$ when $U = \{1, 2, 3, 4\}$ and $\mathcal{F} = \{F, F'\}$ with $F = \{1, 2, 4\}$ and $F' = \{2, 3, 4\}$. For ease of presentation the clique on vertices u_1, \dots, u_4 is not shown; instead sets in the nodes state the allowed colours, where the boldface number is the initial (alpha) colour and the underlined number is the target (beta) colour.

and $b_{F,r-1,y}$: If $j \in U$ is not contained in F then we do not want that a recolouring of v_j in the independent set allows a recolouring in the tree for F . Thus, we use adjacency to the k -clique to forbid the adjacent a - or b -vertex from taking colour 4 (which is exactly the one recolouring option that would have been possible by recolouring v_j). Formally, if j is odd then we make $a_{F,r-1,(j+1)/2}$ adjacent to vertex u_4 of the k -clique (forbidding colour 4 for $a_{F,r-1,(j+1)/2}$), and if j is even then we make $b_{F,r-1,j/2}$ adjacent to u_4 .

This completes the construction of the graph G . We have already specified colours under α and β for all vertices, and we recall that only s and t have different colours with respect to α and β . The necessary recolouring of t to 4, however, will cause a substantial number of recolourings and, along the way, capture the selection of a hitting set for \mathcal{F} . We define the maximum number ℓ of recolouring steps as

$$\ell = 3 + 2p + 2m \cdot 3 \log n.$$

Intuitively, this value is intended as follows: (1) p recolourings for the vertices in the independent set that correspond to a p -hitting set, (2) $m \cdot 3 \log n$ recolourings to propagate the selected hitting set up to all vertices $d_{F,0,1}$ (giving them colour 1), (3) three recolourings for s and t , namely $t \rightarrow 4$, $s \rightarrow 3$, and $t \rightarrow 2$ (4) $m \cdot 3 \log n$ recolourings to undo the hitting set propagation, and (5) p recolourings to undo the selection of the hitting set. We return (G, α, β, ℓ) as the output of our transformation. Since $p < m$ and $\log n = O(m)$, we have that ℓ is polynomially bounded in m , in fact $\ell = O(m^2)$, as claimed. Clearly, our transformation can be performed in polynomial time. It remains to prove correctness.

Completeness. Assume that the initial instance (U, \mathcal{F}, m, p) of HITTING SET(m) is yes and let S a hitting set of size at most p for \mathcal{F} . We outline a recolouring procedure following exactly the stated intuition for the budget ℓ . (All steps are strictly serial but we do not insist on an ordering if it is immaterial.):

1. Recolour all v_j from 4 to 1 for all $j \in S$. The only neighbours are a - or b -vertices that have colours 2 or 3. This uses at most p steps.
2. For each $F \in \mathcal{F}$, recolour bottom-up the vertices in the tree-like claw structure, beginning with some a - or b -vertex whose adjacent independent set vertex has been recoloured from 4 to 1. Since S is a hitting set for \mathcal{F} , such a vertex can always be found.
 - (a) Recolour the a (or b) vertex from 2 (or 3) to 4; there is no conflict with the c -vertex since that has colour 1, same as the adjacent independent set vertex.
 - (b) Recolour the c -vertex from 1 to either 2 or 3; only one choice is possible depending on whether we previously recoloured the a or the b -vertex.
 - (c) Recolour the d -vertex from 4 to 1; there is no conflict with the c -vertex of (now) colour 2 or 3.

At this point the argument can be repeated since the recolouring of, say, $d_{F,x,y}$ with $x \geq 1$, to 1 permits a recolouring of $a_{F,x-1,(y+1)/2}$ or $b_{F,x-1,y/2}$ to 4 depending on the parity of y . Ultimately, we end up with $d_{F,0,1}$ getting colour 1 (which does not conflict with t being colour 3).

Over all sets F this uses $m \cdot 3 \log n$ steps since the tree arrangement has height r and we recolour three vertices in each claw.

3. We then recolour t to 4, s to 3, and t to 2. This costs three steps and fulfills the requirement of β for both vertices. Clearly there are no conflicts.
4. We then trace back the recolourings in each tree structure, using $m \cdot 3 \log n$ steps, followed by undoing the recolourings on vertices v_j corresponding to the hitting set S , using at most p steps. This meets the requirement β for all vertices other than s and t . (Note that $d_{F,0,1}$ changing back from 1 to 4 makes no conflicts with t which now has colour 2.)

Overall we obtain a recolouring sequence of length at most ℓ , as claimed. Thus (G, α, β, ℓ) is indeed yes for k -COLOURING RECONFIGURATION.

Soundness. Let us assume that the obtained instance (G, α, β, ℓ) is yes for k -COLOURING RECONFIGURATION and let $\alpha = \gamma_0, \dots, \gamma_\ell = \beta$ be a sequence of proper recolourings. (We can repeat the last colouring in case that less than ℓ colouring steps are needed.) We begin with some basic arguments about the behaviour of $\gamma_0, \dots, \gamma_\ell$.

If t would never receive colour 4 in any γ_i then it can be easily seen that the sequence must be infeasible: Indeed, this would restrict s to colours 2 and 3, and t to colours 2 and 3. This makes it impossible to, effectively, swap the colours of s and t since they are adjacent. Thus, let $z \in \{1, \dots, \ell\}$ be the smallest integer such that $\gamma_z(t) = 4$. Clearly, since $\beta(t) = 2$ and $\alpha(s) = 2 \neq 3 = \beta(s)$ at least 3 recolouring steps address vertices s and t , leaving at most $2p + 2m \cdot 3 \log n$ for the remaining vertices.

Since $\alpha(d_{F,0,1}) = 4$ for all $F \in \mathcal{F}$ it follows that each $d_{F,0,1}$ must be recoloured to a colour other than 4 before step z (in which t for the first time is coloured 4), and recall that the adjacency to the k -clique forbids all colours other than 1 and 4. As discussed earlier, this propagates the need for earlier recolourings through each tree-like arrangement of claws. Ultimately, for each $F \in \mathcal{F}$ at least one a - or one b -vertex with index $(F, r-1, \cdot)$ must be recoloured before step z , along with a total $3 \log n$ vertices in that tree. Since all these vertices need to be reverted to their original colours later, this leaves only a budget of at most $2p$ for the independent set vertices.

In the independent set, there are forced recolourings from 4 to 1 for all v_j that have adjacent recoloured a - or b -vertices with index $(\cdot, r-1, \cdot)$. Since all these must return to colour 4 in $\gamma_\ell = \beta$ latest, at most p of these vertices can ever be recoloured. Let $S \subseteq U$ denote the elements of U that correspond to these vertices. We will prove that S is a hitting set for \mathcal{F} ; clearly $|S| \leq p$.

Fix any set $F \in \mathcal{F}$. We already argued that a recolouring of $d_{F,0,1}$ from 4 to 1 ultimately requires a prior recolouring of some a - or b -vertex with index $(F, r-1, \cdot)$. Recall, however, that we disallowed such recolourings whenever the corresponding element $j \in U$ is not contained in F . (Here, corresponding refers to our construction where $a_{F,r-1,y}$ is adjacent to v_{2y-1} and $b_{F,r-1,y}$ is adjacent to v_{2y} .) Thus, we must have recoloured an a - or b -vertex with index $(F, r-1, y)$ such that the corresponding element $j \in \{2y-1, 2y\}$ is contained in F . This in turn, as discussed earlier, requires a prior recolouring of v_j which

implies that $j \in S$. Thus, S indeed has a nonempty intersection with F , implying that S is a hitting set for \mathcal{F} , as claimed. \square

5 Conclusions

We have given an affirmative answer for general input graphs to a question posed by Mouawad et al. [26] (IPEC 2013). We showed that RECONFIGURATION is fixed-parameter tractable for k -COLOURING for any fixed $k \geq 1$, when parameterized by the number of recolourings ℓ . It is a natural question to ask whether a single-exponential FPT algorithm can be achieved for this problem. We also proved that the k -COLOURING RECONFIGURATION problem is polynomial-time solvable for $k = 3$, which solves the open problem of Cereceda et al. [11], and that it has no polynomial kernel for all $k \geq 4$, when parameterized by ℓ (up to the standard complexity assumption that $\text{NP} \not\subseteq \text{coNP/poly}$).

References

1. H. L. Bodlaender, B.M.P. Jansen and S. Kratsch, Kernelization lower bounds by cross-composition, *SIAM Journal on Discrete Mathematics* 28 (2014) 277–305.
2. M. Bonamy and N. Bousquet, Recoloring bounded treewidth graphs, *Proc. LAGOS 2013*, *Electronic Notes in Discrete Mathematics* 44 (2013) 257–262.
3. M. Bonamy, M. Johnson, I.M. Lignos, V. Patel and D. Paulusma, Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs, *Journal of Combinatorial Optimization* 27 (2014) 132–143.
4. P. Bonsma, The complexity of rerouting shortest paths, *MFCS 2012*, *Lecture Notes in Computer Science* 7464 (2012) 222–233.
5. P. Bonsma, Independent set reconfiguration in cographs, *Manuscript* (2014) arXiv:1402.1587.
6. P. Bonsma, Rerouting shortest paths in planar graphs, *Proc. FSTTCS 2012*, *LIPIcs* 18 (2012) 337–349.
7. P. Bonsma and L. Cereceda, Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances, *Theoretical Computer Science* 410 (2009) 5215–5226.
8. P. Bonsma, M. Kamiński and M. Wrochna, Reconfiguring independent sets in claw-free graphs, *Manuscript* (2014) arXiv:1403.0359.
9. L. Cereceda, J. van den Heuvel and M. Johnson, Connectedness of the graph of vertex-colourings, *Discrete Mathematics* 308 (2008) 913–919.
10. L. Cereceda, J. van den Heuvel and M. Johnson, Mixing 3-colourings in bipartite graphs, *European Journal of Combinatorics* 30 (2009) 1593–1606.
11. L. Cereceda, J. van den Heuvel and M. Johnson, Finding paths between 3-colourings, *Journal of Graph Theory* 67 (2010) 69–82.
12. R. Diestel, *Graph Theory*. Springer-Verlag, Electronic Edition, 2005.
13. Michael Dom, Daniel Lokshtanov, and Saket Saurabh, Incompressibility through colors and ids, *Proc. ICALP 2009*, *Lecture Notes in Computer Science* 5555 (2009) 378–389.
14. M. R. Garey, D. S. Johnson and L. Stockmeyer, Some simplified NP-complete problems, *Proceedings of the Sixth Annual Symposium on Theory of Computing* (1974) 47–63.

15. P. Gopalan, P. G. Kolaitis, E. N. Maneva and C. H. Papadimitriou, The connectivity of boolean satisfiability: computational and structural dichotomies, *SIAM Journal on Computing* 38 (2009) 2330–2355.
16. J. van den Heuvel, The complexity of change, *Surveys in Combinatorics 2013*, edited by S. R. Blackburn, S. Gerke, and M. Wildon, London Mathematical Society Lecture Notes Series 409.
17. T. Ito, E. D. Demaine, N. J. A. Harvey, C. H. Papadimitriou, M. Sideri, R. Uehara and Y. Uno, On the complexity of reconfiguration problems, *Theoretical Computer Science* 412 (2010) 1054–1065.
18. T. Ito, M. Kamiński and E. D. Demaine, Reconfiguration of list edge-colorings in a graph, *Proc. WADS 2009, Lecture Notes in Computer Science* 5664 (2009) 375–386.
19. T. Ito, K. Kawamura, H. Ono and X. Zhou, Reconfiguration of list $L(2, 1)$ -labelings in a graph, *Proc. ISAAC 2012, Lecture Notes in Computer Science* 7676 (2012) 34–43.
20. T. Ito, K. Kawamura and X. Zhou, An improved sufficient condition for reconfiguration of list edge-colorings in a tree, *Proc. TAMC 2011, Lecture Notes in Computer Science* 6648 (2011) 94–105.
21. T. Ito and E. D. Demaine, Approximability of the subset sum reconfiguration problem, *Proc. TAMC 2011, Lecture Notes in Computer Science* 6648 (2011) 58–69.
22. M. Kamiński, P. Medvedev and M. Milanič, Complexity of independent set reconfigurability problems, *Theoretical Computer Science* 439 (2012) 9–15.
23. M. Kamiński, P. Medvedev and M. Milanič, Shortest paths between shortest paths, *Theoretical Computer Science* 412 (2011) 5205–5210.
24. L. Lovász, Coverings and coloring of hypergraphs, *Proc. 4th Southeastern Conference on Combinatorics, Graph Theory, and Computing, Utilitas Math.* (1973) 3–12.
25. A. E. Mouawad, N. Nishimura and V. Raman, Vertex cover reconfiguration and beyond, *Manuscript* (2014) arXiv:1402.4926.
26. A. E. Mouawad, N. Nishimura, V. Raman, N. Simjour, and A. Suzuki, On the parameterized complexity of reconfiguration problems, *Proc. IPEC 2013, Lecture Notes of Computer Science* 8246 (2013) 281–294.
27. R. Niedermeier, *Invitation to fixed-parameter algorithms*, vol. 31 of Oxford Lecture Series in Mathematics and its Applications, Oxford University Press, Oxford, 2006.